



G06F5/06P

⑪ Publication number:

0 378 195 A3

Doc. Ref. AL2
Appl. No. 09/990,414

EUROPEAN PATENT APPLICATION

⑫ Application number: 90100444.0

⑮ Int. Cl.⁵: G06F 5/06, G06F 13/18

⑬ Date of filing: 10.01.90

⑭ Priority: 10.01.89 JP 3566/89

⑯ Date of publication of application:
18.07.90 Bulletin 90/29

⑰ Designated Contracting States:
DE FR GB

⑱ Date of deferred publication of the search report:
18.03.92 Bulletin 92/12

⑲ Applicant: KABUSHIKI KAISHA TOSHIBA
72, Horikawa-cho Saiwai-ku
Kawasaki-shi Kanagawa-ken 210(JP)

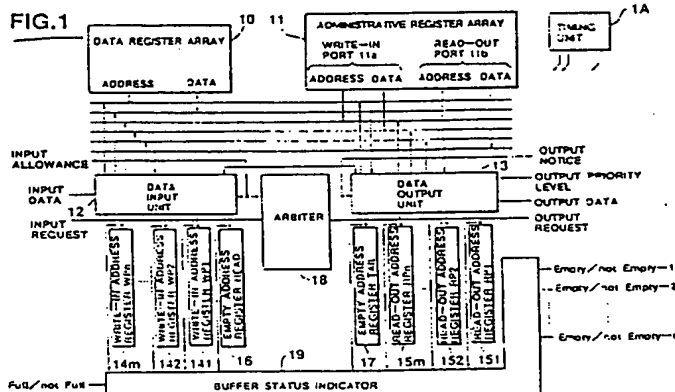
⑳ Inventor: Shobatake, Yasuro
Toshiba-Hiyoshiryo 297,4-11-1,
Hiyoshihoncho
Kohoku-ku, Yokohama-shi,
Kanagawa-ken(JP)
Inventor: Kumaki, Yoshinari
Toshiba-isogodairyo B206, 2-8-2 Shiomidai
Isogo-ku, Yokohama-shi, Kanagawa-ken(JP)

㉑ Representative: Lehn, Werner, Dipl.-Ing. et al
Hoffmann, Eitle & Partner Patentanwälte
Arabellastrasse 4 Postfach 81 04 20
W-8000 München 81(DE)

㉒ Buffer device suitable for asynchronous transfer mode communication.

㉓ A buffer device capable of dealing with multiple priority levels in which the efficiency of the memory capacity utilization can be improved such that the priority levels can be handled at the higher efficiency with smaller memory capacities, and which is adaptable to a high speed buffer implementation. The device includes a data register array (10) containing empty data registers and imaginary FIFO queues, and an administrative register array (11) comprised of a two port RAM (11a,11b) for storing pointer chains specifying the imaginary FIFO queues. The

input of data is accompanied by the modification of the pointer chain to extend it, whereas the output of data is accompanied by the modification of the pointer chain to shorten it, so that the imaginary FIFO queues are administered in flexible manner in order to achieve efficient memory capacity utilization. The procedure for controlling the imaginary FIFO queues can be executed in parallel because of the independency of read and write operations in the two port RAM.





EUROPEAN SEARCH REPORT

EP 90 10 0444

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
D,A	Proceedings of the International Conference on Computer Architecture '88, pp343-354; Tamir et al: 'HIGH-PERFORMANCE MULTI-QUEUE BUFFERS FOR VLSI-COMMUNICATION SWITCH' * abstract *** page 347, paragraph A ** -----	1-4	G 06 F 5/06 G 06 F 13/18
A	IBM TECHNICAL DISCLOSURE BULLETIN. vol. 18, no. 8, January 1976, NEW YORK US pages 2582 - 2583; BOYLE ET AL: 'PACE RECLAMATION IN VIRTUAL STORAGE SYSTEMS' * the whole document ** -----	1	
A	IBM TECHNICAL DISCLOSURE BULLETIN. vol. 24, no. 1A, June 1981, NEW YORK US pages 213 - 215; MACLEAN: 'BUFFER MEMORY ADDRESSING' * the whole document ** -----	1	
A	AFIPS CONFERENCE PROCEEDINGS, vol. 38, 1971, pp601-616, Atlantic City, N.J.; Smith et al: 'SYMBOL- A large experimental system exploring major hardware re- placement of software' * page 606, column 1, line 13 - column 2, line 14; figure 9 ** -----	1	
A	PATENT ABSTRACTS OF JAPAN vol. 13, no. 89 (P-840)8 March 1989 & JP-A-63 279 339 (SANYO) * abstract ** -----	1	G 06 F
A	FR-A-2 559 286 (INRIA) * claim 1 ** -----	1	
The present search report has been drawn up for all claims			
Place of search		Date of completion of search	Examiner
The Hague		17 January 92	COHEN B.
CATEGORY OF CITED DOCUMENTS X: particularly relevant if taken alone Y: particularly relevant if combined with another document of the same category A: technological background O: non-written disclosure P: intermediate document T: theory or principle underlying the invention E: earlier patent document, but published on, or after the filing date D: document cited in the application L: document cited for other reasons &: member of the same patent family, corresponding document			

(12)

EUROPEAN PATENT APPLICATION

(21) Application number: 90100444.0

(51) Int. Cl.⁵: G06F 5/06, G06F 13/18

(22) Date of filing: 10.01.90

(30) Priority: 10.01.89 JP 3566/89

(43) Date of publication of application:
18.07.90 Bulletin 90/29

(84) Designated Contracting States:
DE FR GB

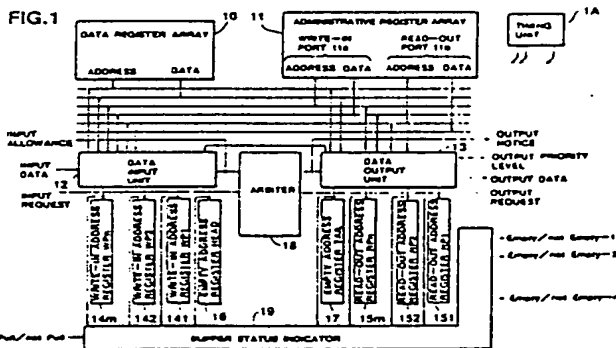
(71) Applicant: KABUSHIKI KAISHA TOSHIBA
72, Horikawa-cho Saiwai-ku
Kawasaki-shi Kanagawa-ken 210(JP)

(72) Inventor: Shobatake, Yasuro
Toshiba-Hiyoshiryo 297,4-11-1,
Hiyoshihoncho
Kohoku-ku, Yokohama-shi,
Kanagawa-ken(JP)
Inventor: Kumaki, Yoshinari
Toshiba-isogodairyo B206, 2-8-2 Shiomidai
Isogo-ku, Yokohama-shi, Kanagawa-ken(JP)

(74) Representative: Lehn, Werner, Dipl.-Ing. et al
Hoffmann, Eitle & Partner Patentanwälte
Arabellastrasse 4 Postfach 81 04 20
D-8000 München 81(DE)

(54) Buffer device suitable for asynchronous transfer mode communication.

(57) A buffer device capable of dealing with multiple priority levels in which the efficiency of the memory capacity utilization can be improved such that the priority levels can be handled at the higher efficiency with smaller memory capacities, and which is adaptable to a high speed buffer implementation. The device includes a data register array (10) containing empty data registers and imaginary FIFO queues, and an administrative register array (11) comprised of a two port RAM (11a,11b) for storing pointer chains specifying the imaginary FIFO queues. The input of data is accompanied by the modification of the pointer chain to extend it, whereas the output of data is accompanied by the modification of the pointer chain to shorten it, so that the imaginary FIFO queues are administered in flexible manner in order to achieve efficient memory capacity utilization. The procedure for controlling the imaginary FIFO queues can be executed in parallel because of the independency of read and write operations in the two port RAM.



plementation.

This object is achieved in the present invention by providing a buffer device for receiving, temporarily storing and transmitting data accompanied by information indicating a priority level of the data, where there is at least two distinct priority levels, the device comprising: data register array including a plurality of data register means for temporarily storing the data, the data register means being divided into empty data register means storing no data and as many number of imaginary FIFO queues as a number of distinct priority levels, each imaginary FIFO queue being corresponding to each distinct priority level, and a number of data register means in each imaginary FIFO queue being flexible; administrative register array comprised of a two port RAM means for modifiably storing the administrative information containing pointer chains specifying the imaginary FIFO queues in the data register array, having a write-in port through which the administrative information can be written into the administrative register array and a read-out port through which the administrative information can be read out from the administrative register array which are independently operable; data input means for receiving new data, entering the new data into one empty data register means, and modifying the pointer chain specifying the imaginary FIFO queue corresponding to a priority level of the new data indicated by the information accompanying the new data, such that the pointer chain is extended to include that one empty data register means into which the new data is entered at an end of that imaginary FIFO queue; and data output means for taking out data stored in one data register means, transmitting the data, and modifying the pointer chain specifying the imaginary FIFO queue corresponding to a priority level of that data, such that the pointer chain is shortened to exclude that one data register means from which the data is taken from a top of that imaginary FIFO queue.

Other features and advantages of the present invention will become apparent from the following description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of one embodiment of a buffer device according to the present invention.

Fig. 2 is a detailed block diagram of an administrative register array of the buffer device of Fig. 1.

Fig. 3 is a flow chart for the input of input data in the buffer device of Fig. 1.

Fig. 4 is a flow chart for the output of output

data in the buffer device of Fig. 1.

Fig. 5 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining their functions.

Fig. 6 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining their functions in the input of new data for a general case.

Fig. 7 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining the result of the input of new data explained in Fig. 6.

Fig. 8 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining their functions in the input of new data for one special case.

Fig. 9 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining the result of the input of new data explained in Fig. 8.

Fig. 10 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining their functions in the input of new data for another special case.

Fig. 11 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining the result of the input of new data explained in Fig. 10.

Fig. 12 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining their functions in the output of data for a general case.

Fig. 13 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining the result of the output of data explained in Fig. 12.

Fig. 14 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining their functions in the output of data for one special case.

Fig. 15 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining the result of the output of data explained in Fig. 14.

Fig. 16 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining their functions in the output of data for another special case.

Fig. 17 is a diagram showing imaginary FIFO queues, empty data registers, and pointer chains in the buffer device of Fig. 1 for explaining the result of the output of data explained in Fig. 16.

Fig. 18 is a diagram showing imaginary FIFO

read-out port 11b, by utilizing the empty address register tail 17 and one of read-out address registers 151-15m which is selected in accordance with a bit pattern given externally which indicates the priority level of the output data, as before.

The arbiter 18 arbitrates conflicting requests when an input request and an output request occurred simultaneously by selecting one of the input request and the output request. In the following description of this embodiment, it is assumed that the output request is always given a higher priority than the input request, though this can be reversed if desired.

The buffer status indicator 19 gives out a buffer full signal for indicating whether the buffer device has any memory capacity for accepting new data available, by comparing the bit patterns stored in the empty Address register head 16 and empty address register tail 17.

In addition, the buffer status indicator 19 gives out buffer empty signal for indicating whether there is any data for output in each of imaginary FIFO queues formed inside the data register array 10 in correspondence with each priority level, by comparing the bit patterns stored in each one of the write-in address registers 141-14m corresponding to each priority level and each one of the read-out address registers 151-15m corresponding to each priority level.

The timing unit 1A provides pulse sequences indicating timings necessary to activate elements of the buffer device such as the write-in pulse signal and the clear pulse signal utilized in the administrative register array 11 described above.

In this embodiment, the input data are entered into the buffer device in accordance with the flow chart of Fig. 3, as follows. Namely, first whether the buffer full signal from the buffer status indicator 19 indicates full or not full is determined(step 101). If full, the input of the input data is postponed until the buffer full signal changes to not full. Otherwise, the memory capacity for accepting new data is available(step 102) so that the input request is given to the arbiter 18(step 103). Only when the input allowance is given from the arbiter 18(step 104), the input data are entered into the data input unit 12(step 105) and then the input data are written into the data register array 10 by the data input unit 12(step 106), which completes the input of the input data.

Also, in this embodiment, the output data are sent out from the buffer device in accordance with the flow chart of Fig. 4, as follows. Namely, first whether the buffer empty signal from the buffer status indicator 19 corresponding to the imaginary FIFO queue for a desired priority level for next output data indicates empty or not empty is determined(step 201). If empty, the output of the

output data is postponed until the buffer empty signal changes to not empty. Otherwise, the data to be output data are present(step 202) so that the output priority level is given to the data output unit 13(step 203) and the output request is given to the arbiter 18(step 204). Only when the output notice is given from the arbiter 18(step 205), the output data are read from the data register array 10 by the data output unit 13 (step 206) and then the output data are sent out from the data output unit 13(step 207), which completes the output of the output data.

Referring now to Figs. 5 to 18, a manner of setting up imaginary FIFO queues in the data register array 10 in this embodiment will be explained in detail.

In Figs. 5 to 18, HEAD stands for the empty address register head 16, TAIL stands for the empty address register tail 17, RPI stands for the read-out address register 151, RP2 stands for the write-in address register 152, WP1 stands for the write-in address register 141, WP2 stands for the write-in address register 142. In addition, a data register having a certain address in the data register array 10 and a corresponding administrative register having the same certain address in the administrative register array 11 are depicted as a pair, and arrows joining the administrative registers represents pointer chains established between the administrative registers.

Each of the administrative registers, read-out address registers, write-in address registers, empty address register head, and empty address register tail has a capacity for storing an address associated with each data register, so as to furnish a function of pointer.

An address consisting binary zeros alone is reserved for an indication of the absence of the pair of data register and administrative register to point to, and is represented by a slash in a box for administrative registers, read-out address registers, write-in address registers, empty address register head, and empty address register tail.

Here, for the sake of simplicity, the explanation will be given for a case involving only two pairs of read-out register and two write-in register in which two imaginary FIFO queues are to be constructed on the data register array 10, in which case there are two sets of pointer chains provided.

Fig. 5 shows a state of the buffer device diagrammatically. Two imaginary FIFO queues are represented by boxes joined by two pointer chains starting from the read-out registers RP1 and RP2, and the last data register and administrative register pairs of the imaginary FIFO queues each of which has a slash in a box for the administrative register are pointed by the write-in registers WP1 and WP2. The administrative registers associated with the empty data registers are also joined by a

output of the output data.

One of such cases is that in which the imaginary FIFO queue for a particular priority level becomes empty as a result of this output of the output data, as shown in Fig. 14.

In this case, the procedure for the output of the output data differs from that described above with Fig. 14 at the step indicated by a chain line H. Namely, in this case the address stored in the administrative register associated with that data is transferred not only to the read-out address register RP1 for that imaginary FIFO queue of that particular priority level, as indicated by a chain line H in Fig. 14, but also to the write-in address register WP1 of the same imaginary FIFO queue, as indicated by a chain line I in Fig. 14, so that after this output of the output data both the read-out address register RP1 and write-in address register WP1 have the address consisting of binary zeros alone, as shown in Fig. 15. This will subsequently be detected by the buffer status indicator 19 which produces the buffer empty signal for that particular priority level in response.

Another case calling for the special operation is that in which all the data registers are filled up before this output of the output data such that both of the empty address register head HEAD and empty address register tail TAIL stores the address consisting of binary zeros alone, as shown in Fig. 16. This situation will be detected by the buffer status indicator 19 which produces the buffer full signal in response.

In this case, the procedure for the output of the output data differs from that described above with Fig. 16 at the step indicated by a chain line J. Namely, in this case there is no administrative register pointed by that empty address register tail TAIL to transfer the original address stored in the read-out address register RP2 to. Thus, in this case, the original address stored in the read-out address register RP2 is transferred to the empty address register tail TAIL, as indicated by a chain line J in Fig. 16, as well as to the empty address register head HEAD, as indicated by a chain line K in Fig. 16, so that after this output of the output data both the empty address register head HEAD and the empty address register tail TAIL point to the same data register from which the output data has been taken, as shown in Fig. 17.

Fig. 18 shows a state of the buffer device in which all the data registers are empty. In this case, all pairs of the data registers and the administrative registers are joined by a pointer chain starting from the empty address register head HEAD and ending at the empty address register tail TAIL, while the read-out address registers RP1 and RP2, as well as the write-in address registers WP1 and WP2

store the address consisting of binary zeros alone.

It is to be noted that in these occasions of input and output explained above, the various steps involved can be carried out simultaneously, since the administrative register array 11 has a two port RAM structure involving the write-in port 11a and read-out port 11b which can be operated independently. Therefore, this buffer device is suitable for a high speed buffer implementation.

When this buffer device is applied to the packet exchange system or communication system using ATM, the data to be stored in the data register array 10 will be the packets or cells, respectively.

In such a case, the priority levels may designate the direction of transmission of the packets or cells such that the buffer device can be viewed as a packet switch or a cell switch. Moreover, in a communication system with ATM, when a plurality of input paths to the buffer device are provided and the directions of cells are taken for the priority levels, this buffer device can function as a type of a cell switch known as a shared buffering cell switch.

As explained, it is possible to provide a buffer device capable of dealing with multiple priority levels in which the efficiency of the memory capacity utilization can be improved such that the priority levels can be handled at the higher efficiency with smaller memory capacities, and which is adaptable to a high speed buffer implementation, because each of the imaginary FIFO queues corresponding to different priority levels has flexible memory capacity, so that even when the data for a particular priority level are more numerous than those for the other priority data, a memory capacities of the entire buffer structure will be utilized at high efficiency, and the procedure for controlling the imaginary FIFO queues can be executed in parallel.

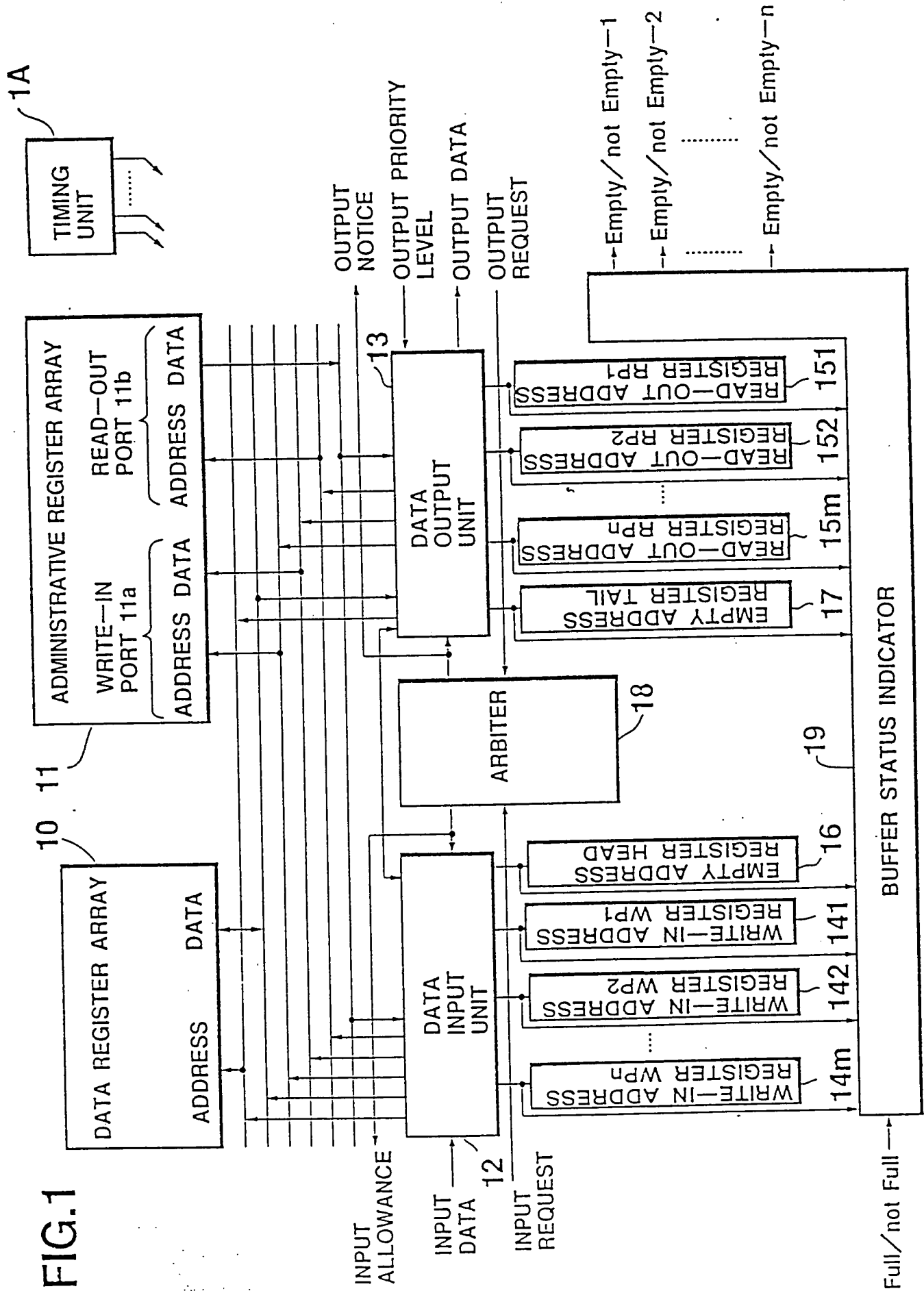
Besides those already mentioned above, many modifications and variations of the above embodiments may be made without departing from the novel and advantageous features of the present invention. Accordingly, all such modifications and variations are intended to be included within the scope of the appended claims.

Reference signs in the claims are intended for better understanding and shall not limit the scope.

Claims

1. A buffer device for receiving, temporarily storing and transmitting data accompanied by information indicating a priority level of the data, where there is at least two distinct priority levels, the device comprising:
data register array(10) including a plurality of data register means for temporarily storing the data, the data register means being divided into empty data

FIG. 1



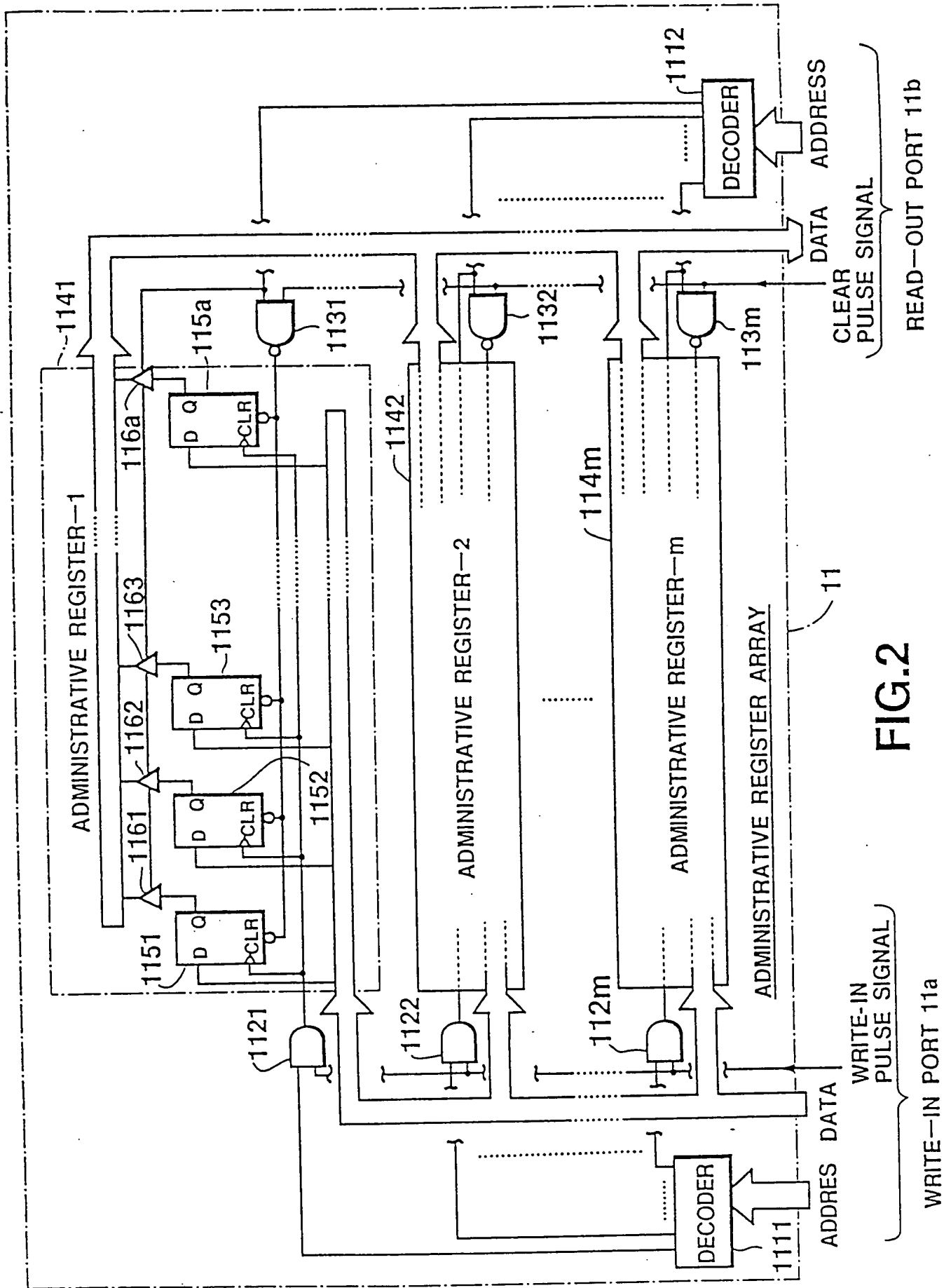


FIG.2

FIG.3

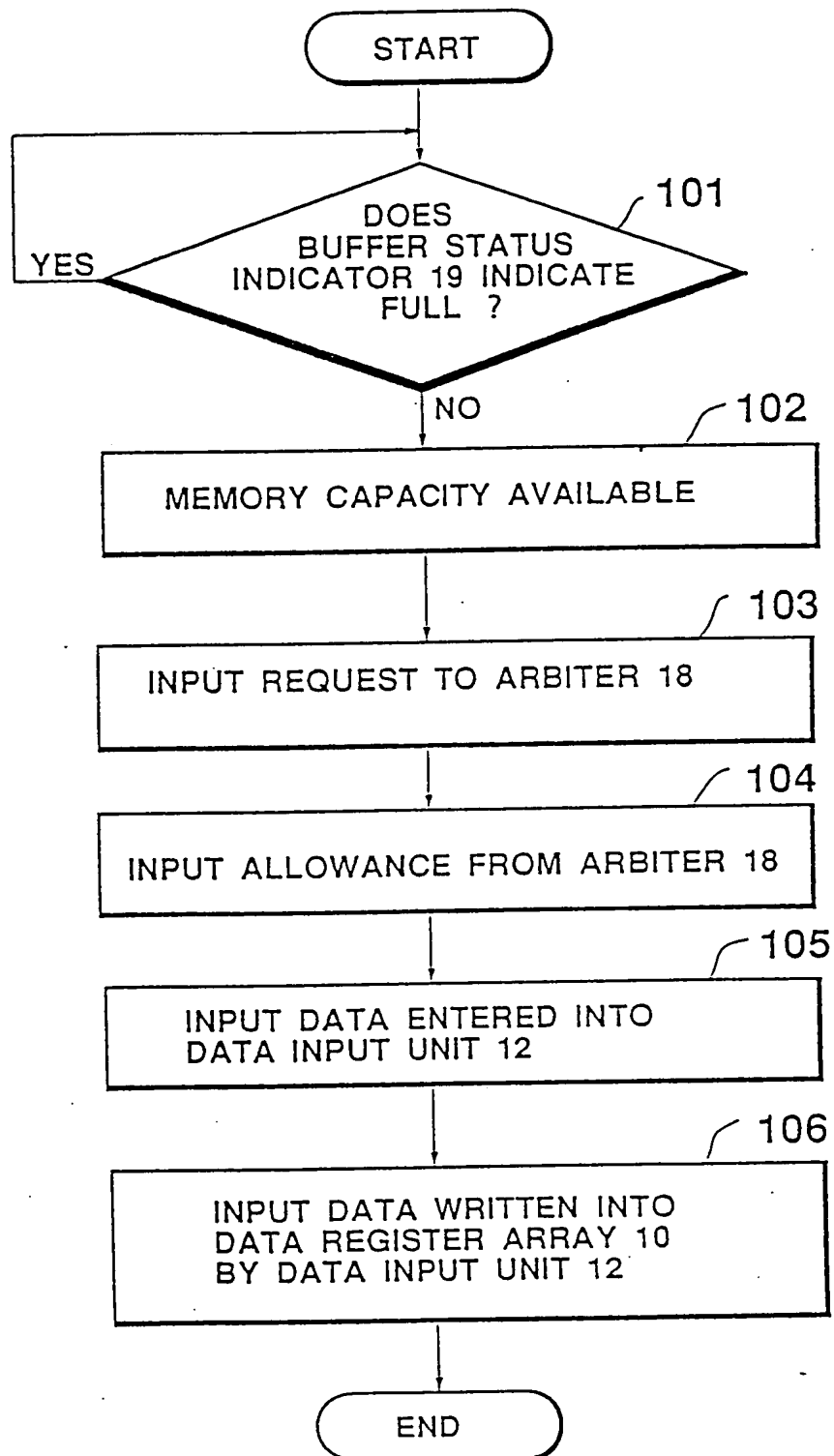


FIG.4

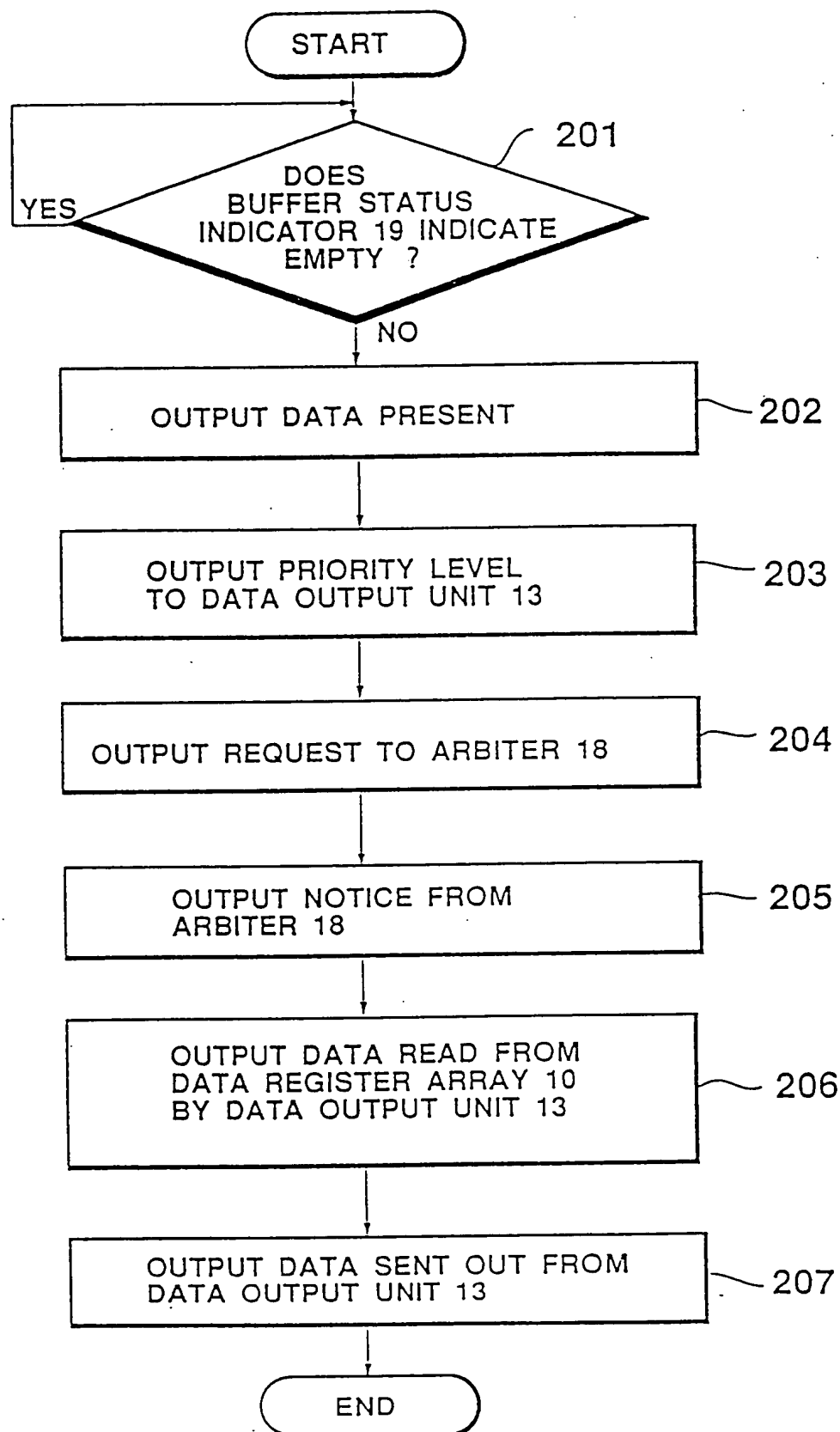
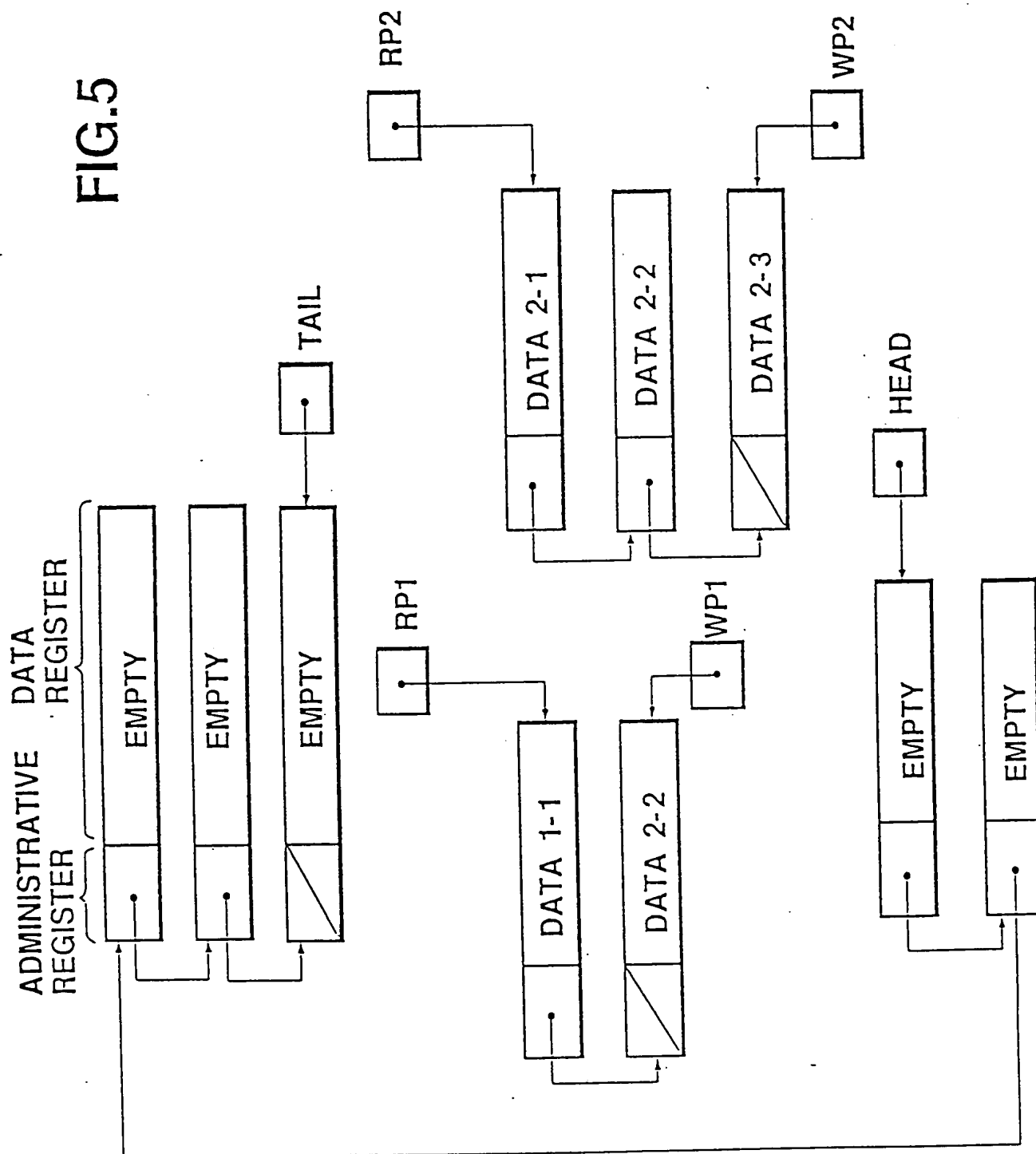


FIG.5



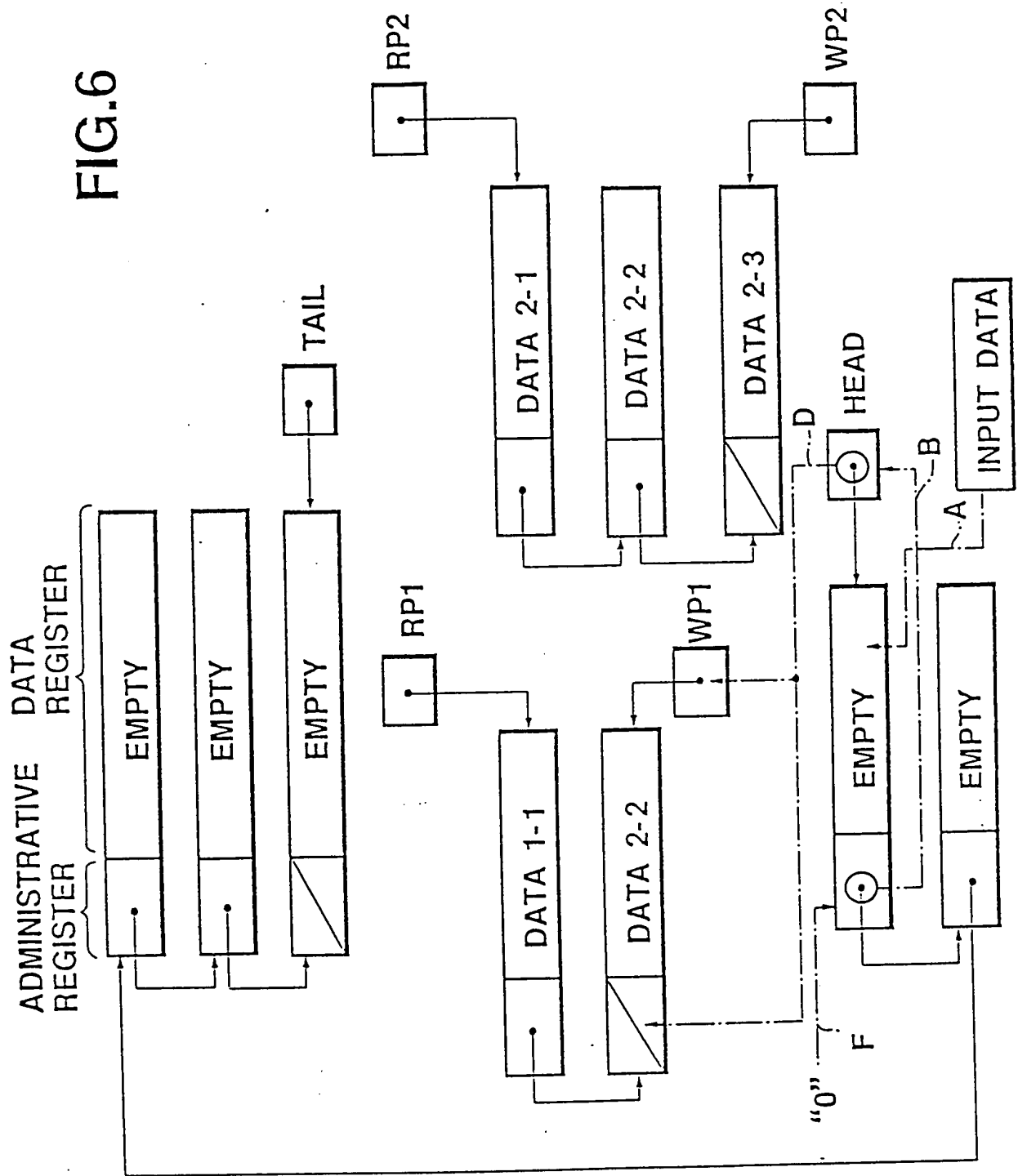


FIG. 7

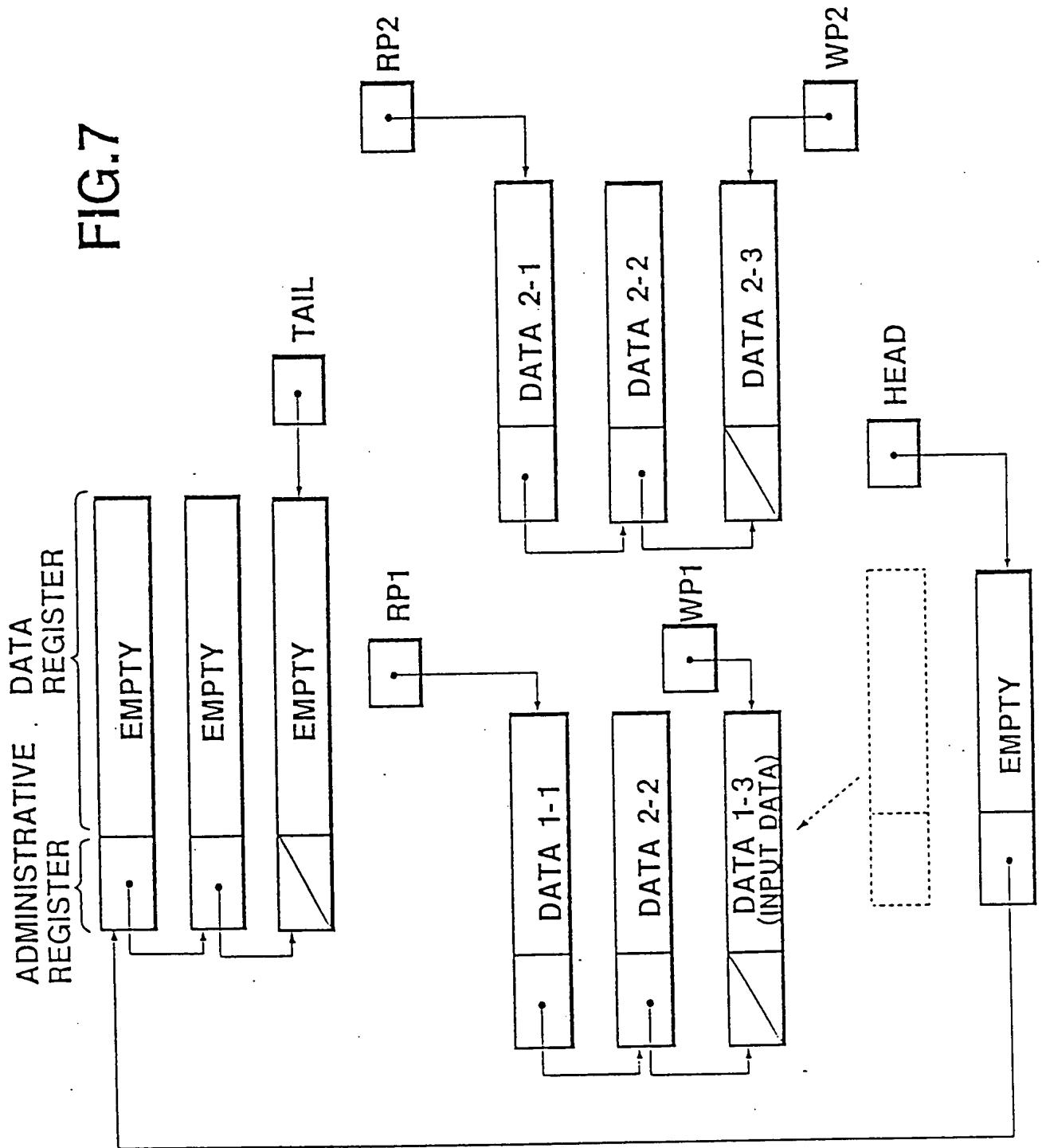


FIG.8

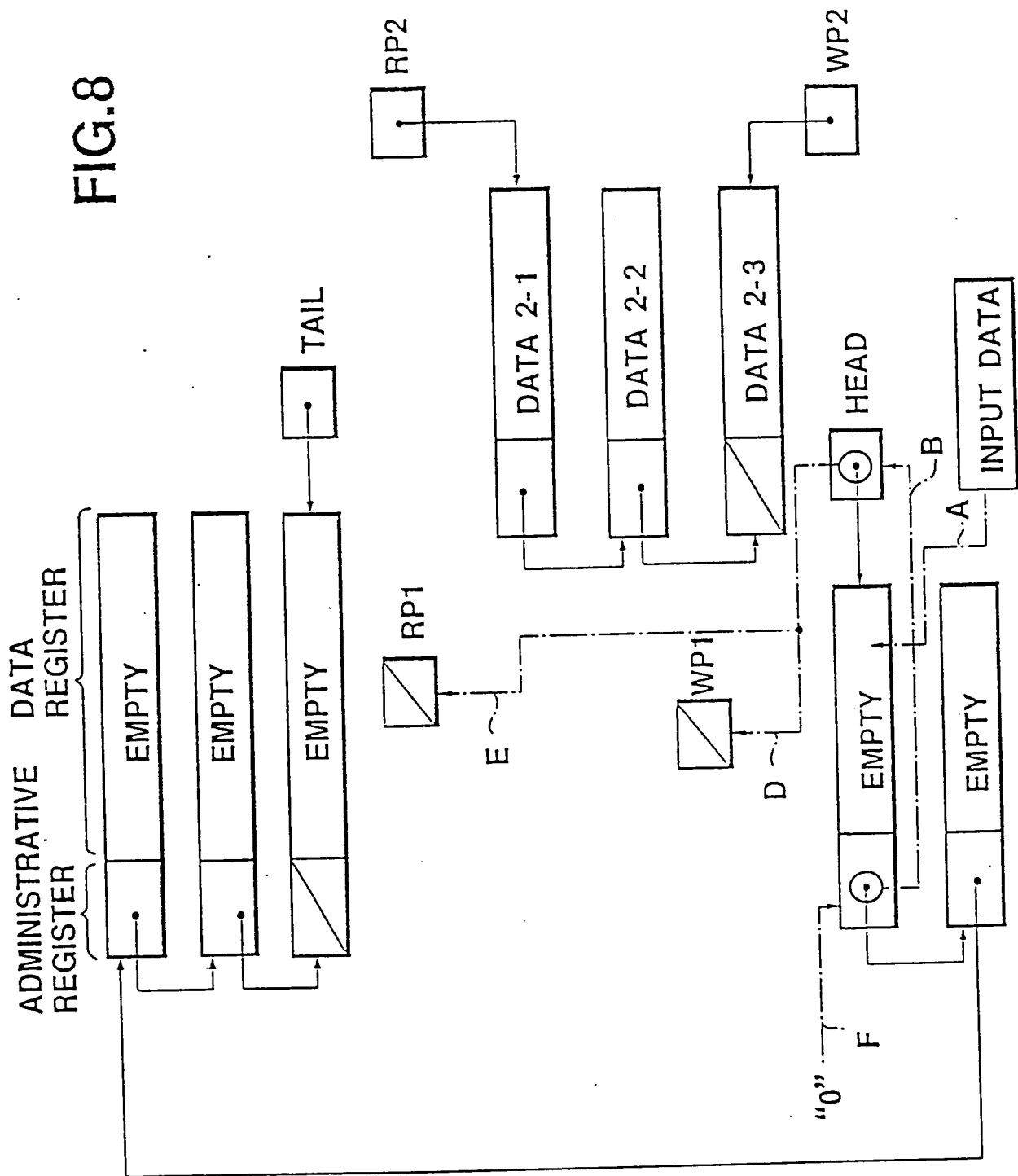


FIG. 9

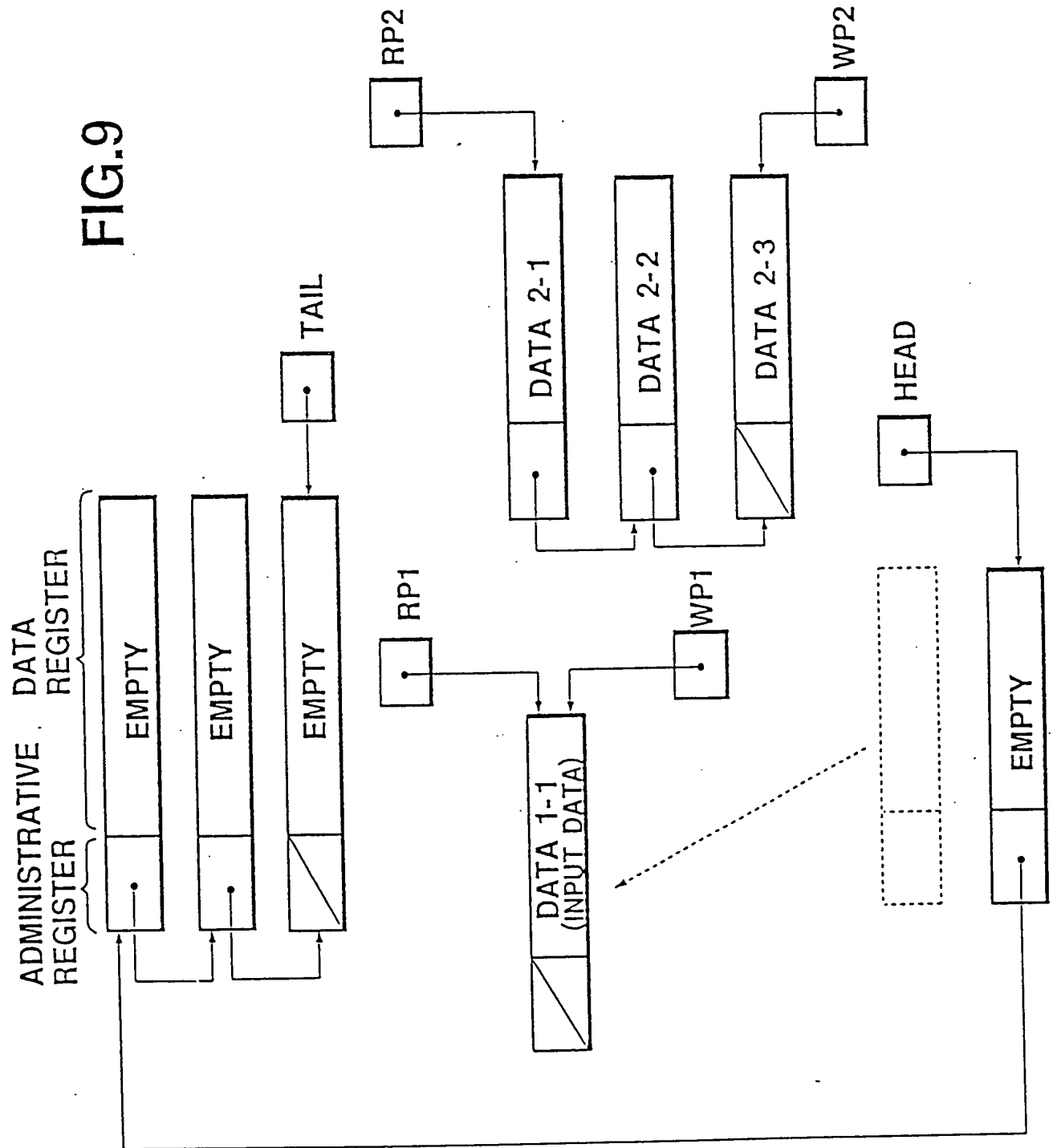


FIG. 10

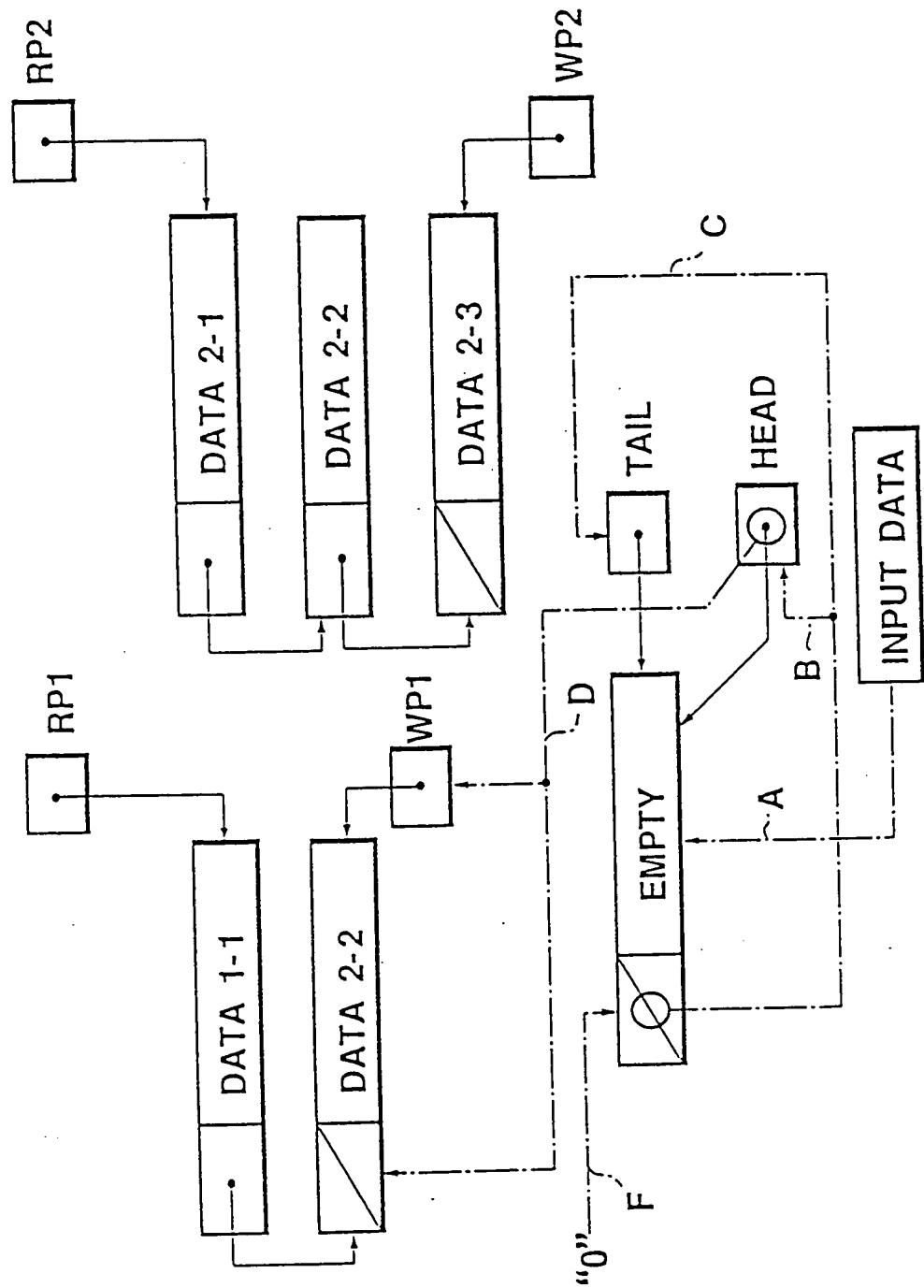
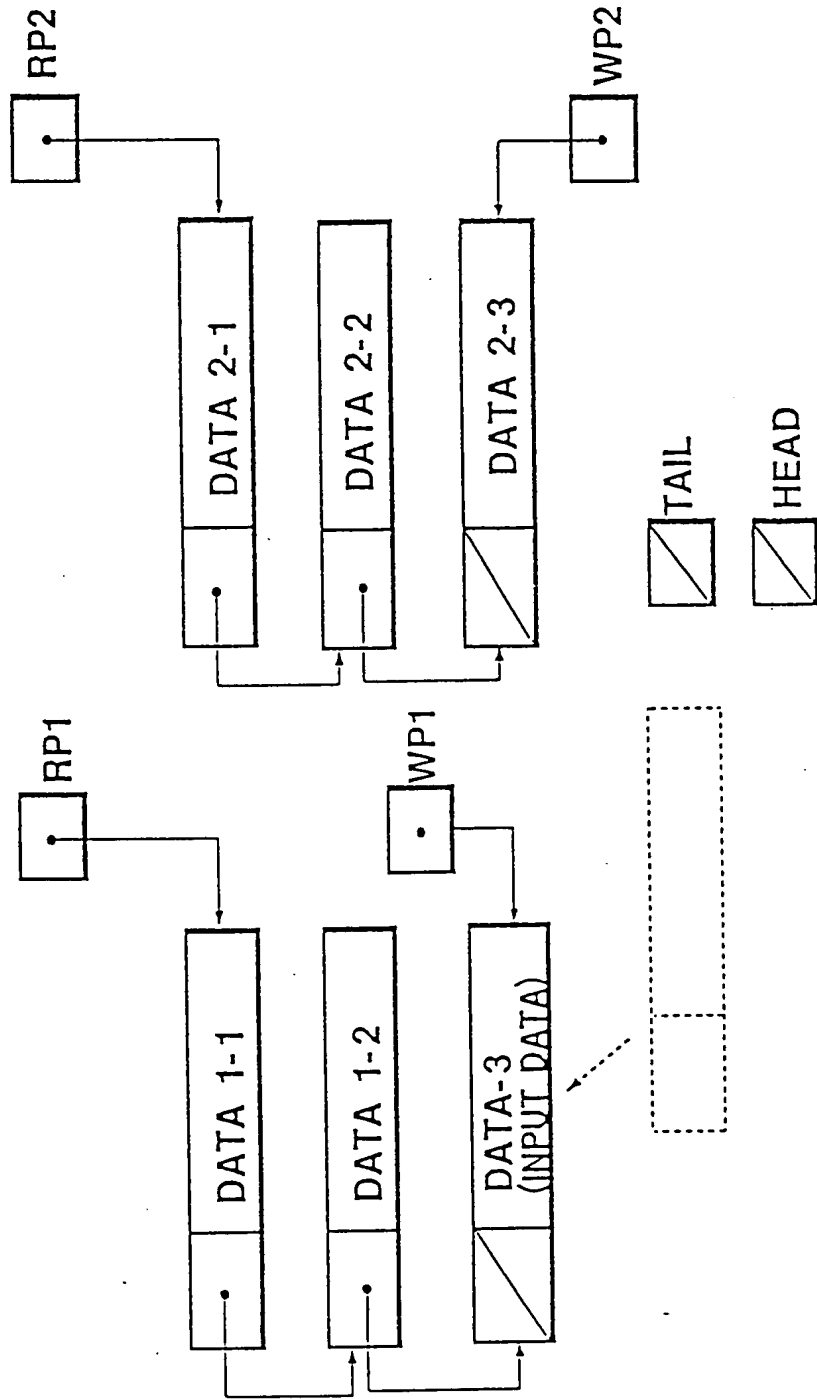


FIG.11



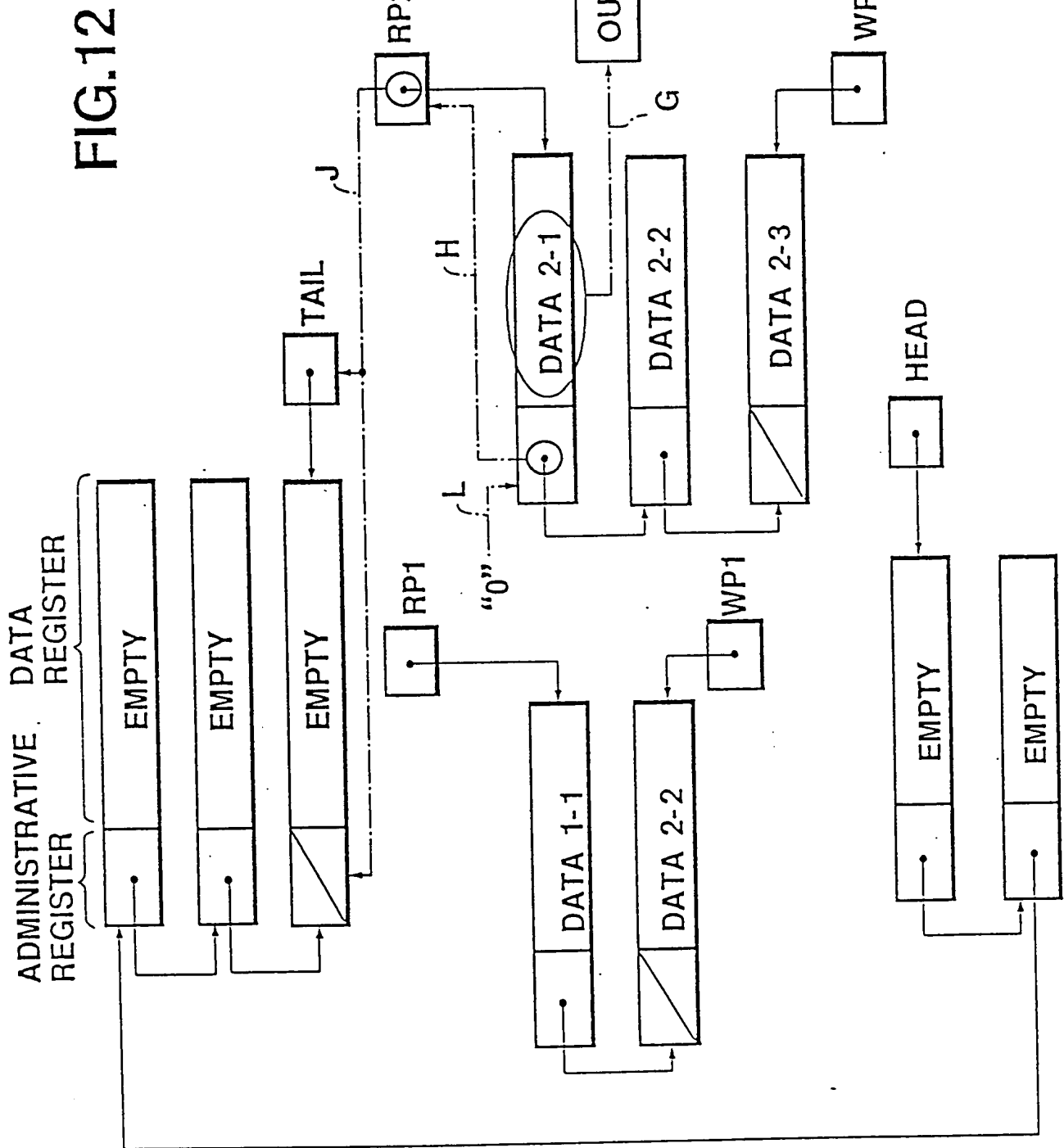


FIG. 13

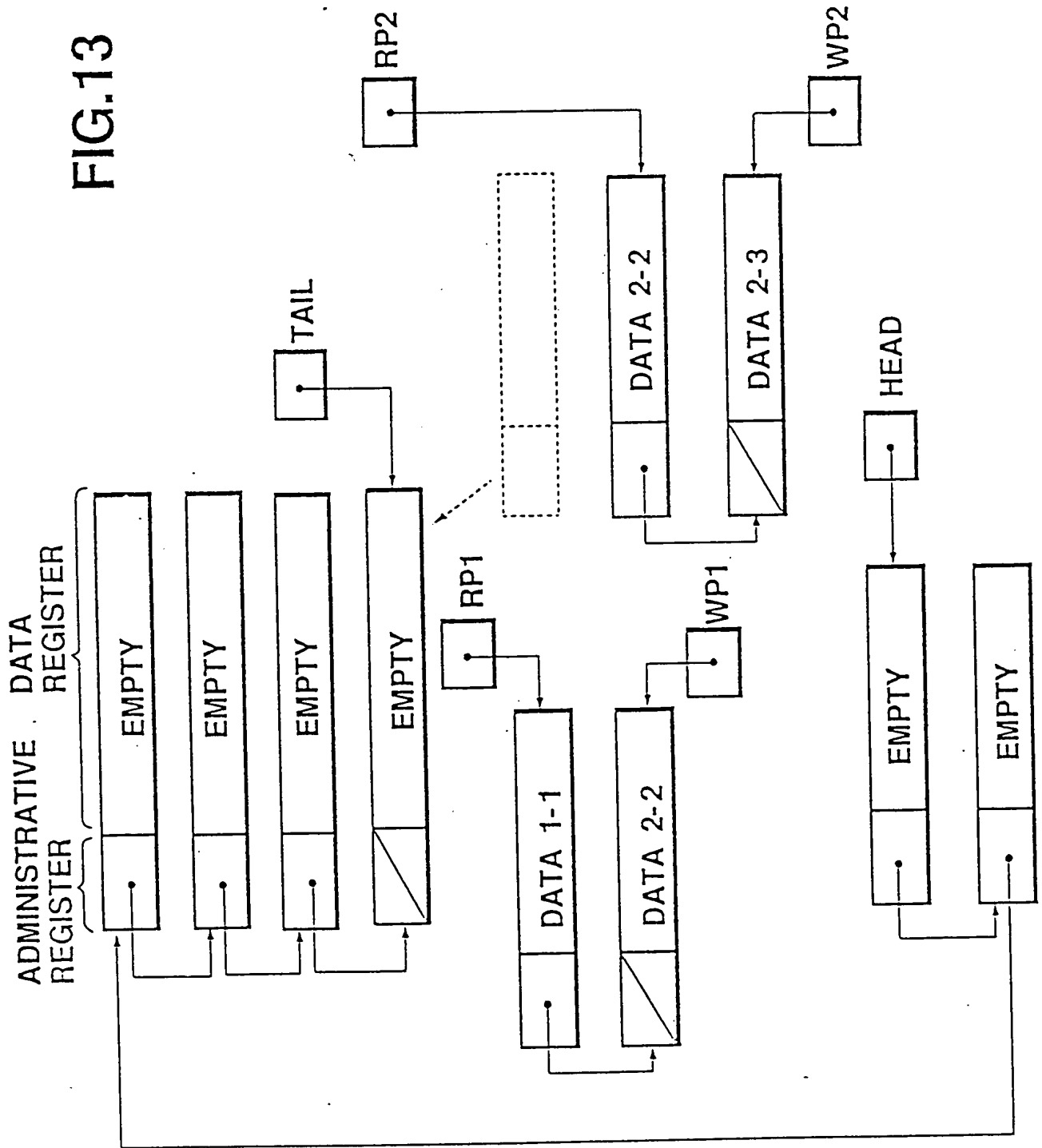
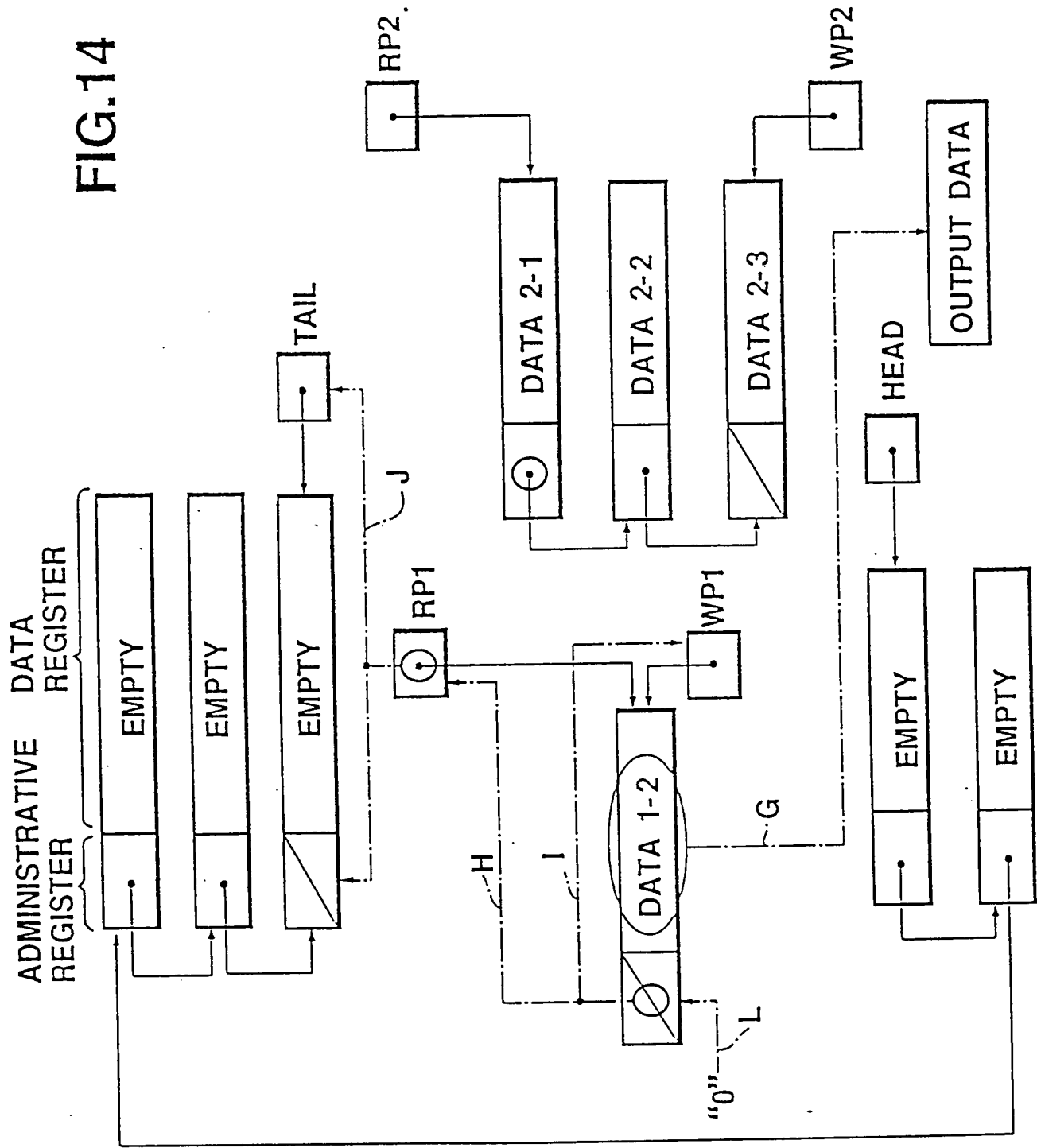


FIG. 14



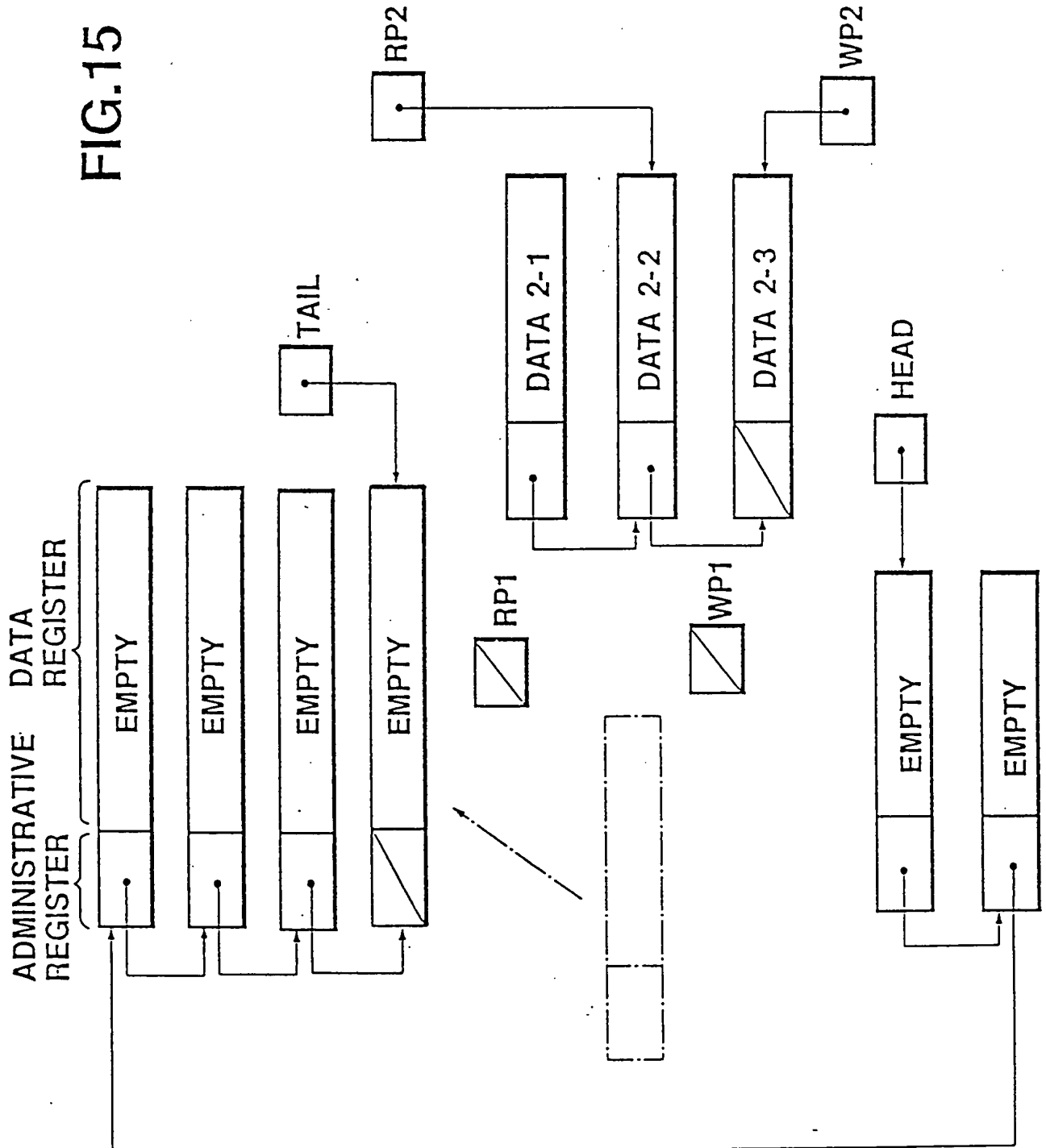


FIG.16

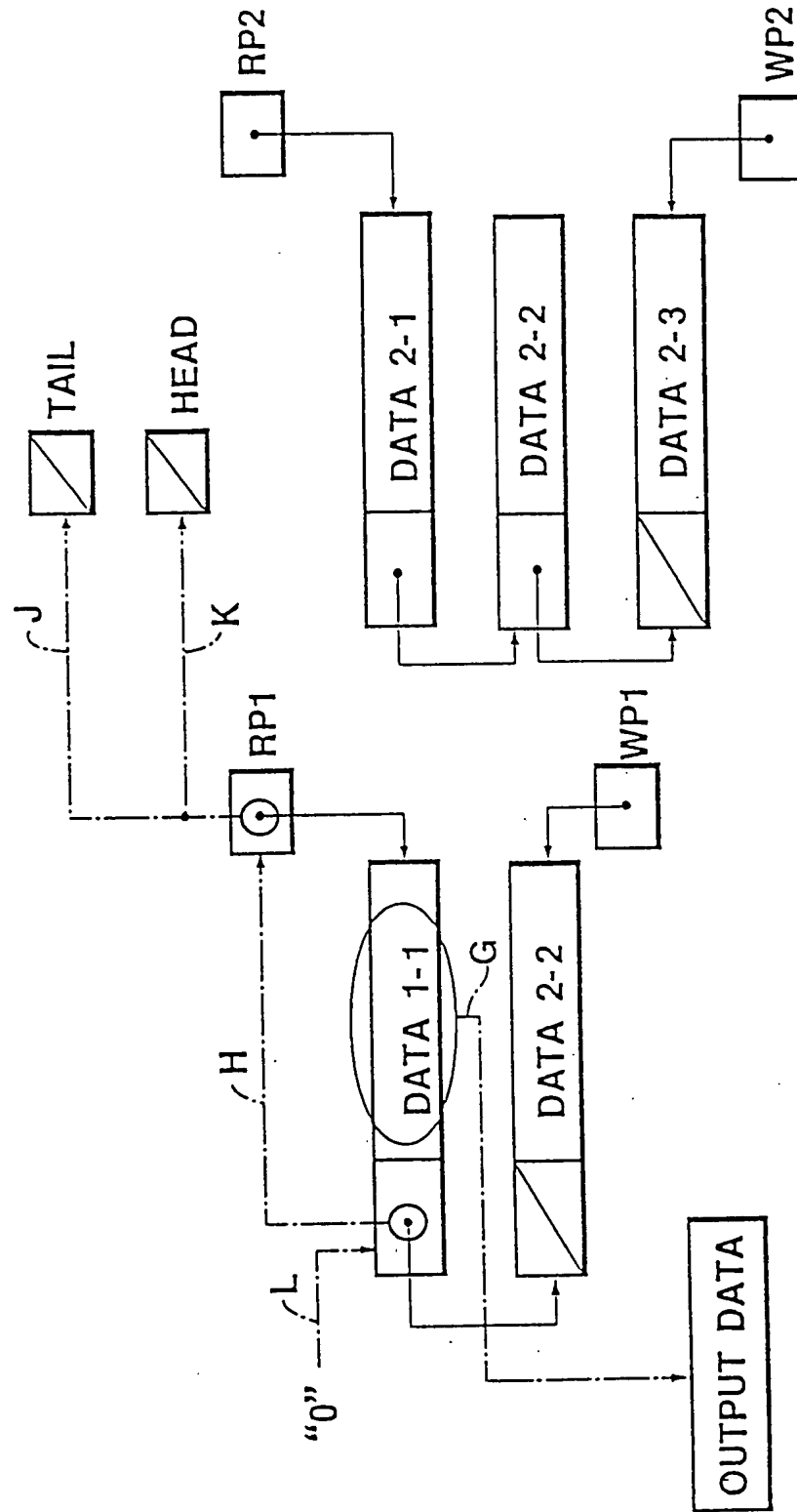


FIG.17

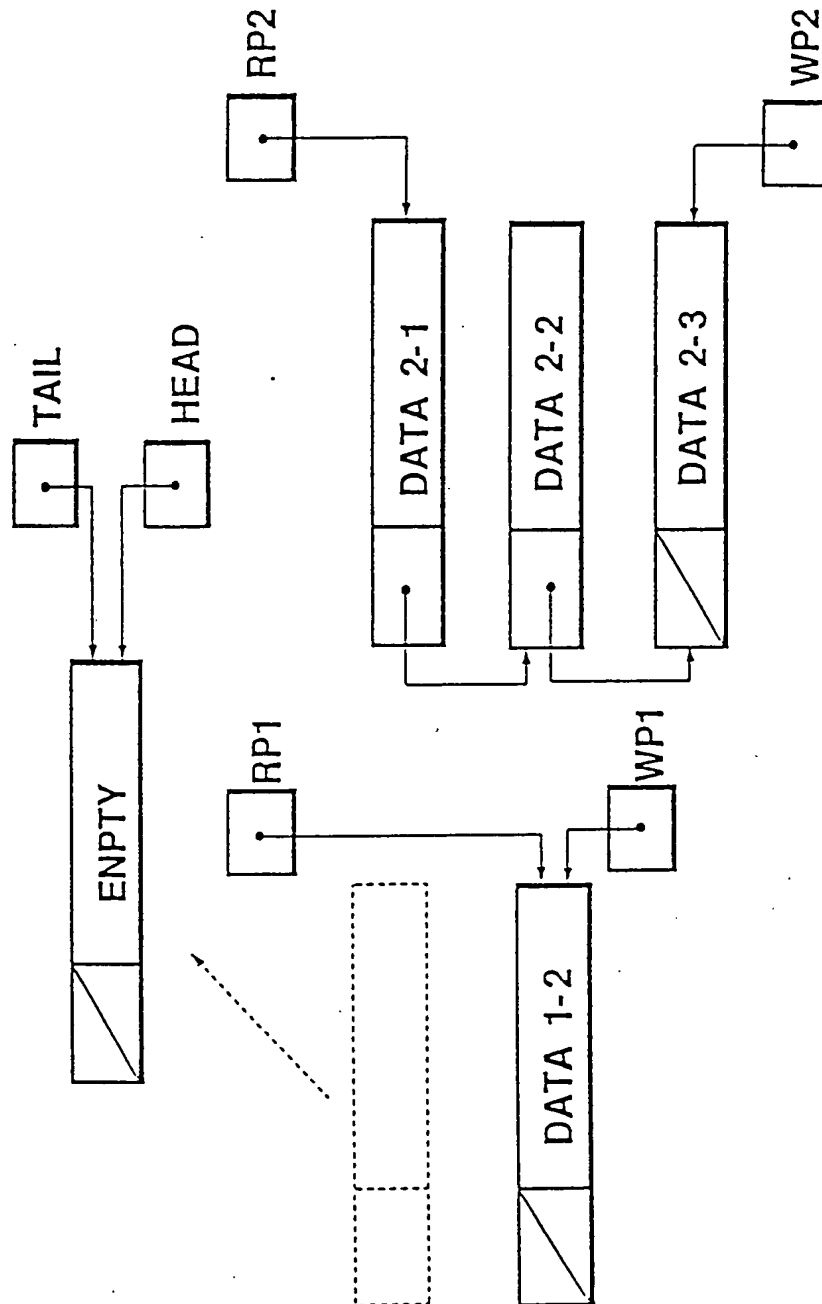


FIG.18

